Amendments to the Claims:

Please add new claims 37-42 as shown in the LISTING OF CLAIMS below. Please amend claims 5-7, 29-32, and 35-36 as shown in the LISTING OF CLAIMS below. Claims 1-4, and 26-28 remain unchanged. The LISTING OF CLAIMS will replace all prior versions, and listings, of claims in the present application.

Listing of Claims:

1. (original) A method for executing a dynamically loaded program, said program including a main program unit, said method comprising:

executing said main program unit a first time;

creating at least one library file containing only application program files loaded during said first execution of said main program unit;

specifying a system program file input; and

executing said main program unit a second time using said system program file input and said at least one library file for dynamically loaded program files.

2. (original) The method of claim 1 wherein said creating further comprises:

specifying a first at least one pathname for system program files;

specifying a second at least one pathname for application program files;

executing said main program unit using said first at least one pathname and said second at least one pathname for dynamically loaded program files; and

storing each application program file loaded during execution of said main program unit to said library file.

3. (original) The method of claim 2 wherein said storing further comprises:

loading a program file when referenced during execution of said main program unit;

storing said program file to said library file when said program file is an application program file; and

determining whether execution of said main program unit has terminated.

4. (original) The method of claim 3 wherein said library file further comprises a compressed file.

5. (currently amended) The method of claim 1 wherein said program files comprise ~~Java™~~ class files and ~~Java™~~ archive files.

6. (currently amended) The method of claim 5 wherein said system program file input comprises ~~the Java™ Development Kit~~ a collection of services and libraries included in an execution environment.

7. (currently amended) The method of claim 6 wherein said library file comprises an ~~Java™~~ archive file.

8. (withdrawn) A method for optimizing a dynamically loaded program, said program including a main program unit, said method comprising:

creating at least one library file containing only application program files loaded during execution of said main program unit; and

optimizing said program based upon a list of application program files in said library file.

9. (withdrawn) The method of claim 8 wherein said creating further comprises:

specifying a first at least one pathname for system program files;

specifying a second at least one pathname for application program files;

executing said main program unit using said first at least one pathname and said second at least one pathname for dynamically loaded program files; and

storing each application program file loaded during execution of said main program unit to said library file.

10. (withdrawn) The method of claim 9 wherein said storing further comprises:

loading a program file when referenced during execution of said main program unit;

storing said program file to said library file when said program file is an application program file; and

determining whether execution of said main program unit has terminated.

11. (withdrawn) The method of claim 10 wherein said library file further comprises a compressed file.

12. (withdrawn) The method of claim 8 wherein said program files comprise Java™ class files and Java™ archive files.

13. (withdrawn) The method of claim 12 wherein said system program file input comprises the Java™ Development Kit.

14. (withdrawn) The method of claim 13 wherein said library file comprises a Java™ archive file.

15. (withdrawn) The method of claim 13 wherein said optimizing further comprises:

receiving an application program pathname of a referenced program unit in said library file;

determining a referencing program unit when said pathname is unexpected, said referencing program unit referencing said referenced program unit; and

modifying said referencing program unit to remove any reference to said referenced unit.

16. (withdrawn) A method for testing a dynamically loaded program, said program including a main program unit, said method comprising:

specifying a list including at least one application program file to be tested;

creating at least one library file containing only application program files loaded during execution of said main program unit; and

indicating incomplete test coverage when at least one file in said list is not represented in said library file.

17. (withdrawn) The method of claim 16 wherein said creating further comprises:

specifying a first at least one pathname for system program files;

specifying a second at least one pathname for application program files;

executing said main program unit using said first at least one pathname and said second at least one pathname for dynamically loaded program files; and

storing each application program file loaded during execution of said main program unit to said library file.

18. (withdrawn) The method of claim 17 wherein said storing further comprises:

loading a program file when referenced during execution of said main program unit;

storing said program file to said library file when said program file is an application program file; and

determining whether execution of said main program unit has terminated.

19. (withdrawn) The method of claim 18 wherein said library file further comprises a compressed file.

20. (withdrawn) The method of claim 16 wherein said program files comprise Java™ class files and Java™ archive files.

21. (withdrawn) The method of claim 20 wherein said system program file input comprises the Java™ Development Kit.

22. (withdrawn) The method of claim 21 wherein said library file comprises a Java™ archive file.

23. (original) An apparatus for executing a dynamically loaded program, said program including a main program unit, said apparatus comprising:

means for executing said main program unit a first time;

means for creating at least one library file containing only application program files loaded during said first execution of said main program unit;

means for specifying a system program file input; and

means for executing said main program unit a second time using said system program file input and said at least one library file for dynamically loaded program files.

24. (withdrawn) An apparatus for optimizing a dynamically loaded program, said program including a main program unit, said apparatus comprising:

means for creating at least one library file containing only application program files loaded during execution of said main program unit; and

means for optimizing said program based upon a list of application program files in said library file.

25. (withdrawn) An apparatus for testing a dynamically loaded program, said program including a main program unit, said apparatus comprising:

means for specifying a list including at least one application program file to be tested;

means for creating at least one library file containing only application program files loaded during execution of said main program unit; and means for indicating incomplete test coverage when at least one file in said list is not represented in said library file.

26. (original) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method to execute a dynamically loaded program, the method comprising:

executing said main program unit a first time;

creating at least one library file containing only application program files loaded during said first execution of said main program unit;

specifying a system program file input; and

executing said main program unit a second time using said system program file input and said at least one library file for dynamically loaded program files.

27. (original) The program storage device of claim 26 wherein said creating further comprises:

specifying a first at least one pathname for system program files;

specifying a second at least one pathname for application program files;

executing said main program unit using said first at least one pathname and said second at least one pathname for dynamically loaded program files; and

storing each application program file loaded during execution of said main program unit to said library file.

28. (original) The program storage device of claim 27 wherein said storing further comprises:

loading a program file when referenced during execution of said main program unit;

storing said program file to said library file when said program file is an application program file; and

determining whether execution of said main program unit has terminated.

29. (currently amended) The program storage device of claim ~~26~~ 28 wherein said library file further comprises a compressed file.

30. (currently amended) The program storage device of claim 26 wherein said program files comprise ~~Java™~~ class files and ~~Java™~~ archive files.

31. (currently amended) The program storage device of claim 30 wherein said system program file input comprises ~~Java™~~ class files and ~~Java™~~ archive files.

32. (currently amended) The program storage device of claim 31 wherein said library file comprises an ~~Java™~~ archive file.

33. (withdrawn) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method to optimize a dynamically loaded program, the method comprising:

creating at least one library file containing only application program files loaded during execution of said main program unit; and

optimizing said program based upon a list of application program files in said library file.

34. (withdrawn) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method to test a dynamically loaded program, the method comprising:

specifying a list including at least one application program file to be tested;

creating at least one library file containing only application program files loaded during execution of said main program unit; and

indicating incomplete test coverage when at least one file in said list is not represented in said library file.

35. (currently amended) A method for representing a library file, said method including:

storing in at least one program unit field the pathname of every application program unit loaded during the execution of a dynamically loaded program, said dynamically loaded program including a main program unit; and

storing in a main unit field the pathname of said main program unit.

36. (currently amended) The method of claim 35 wherein

said program unit field and said main unit field are contained within ~~a JAR file~~ an archive file; and

said main unit field comprises a manifest file.

37. (new) The apparatus of claim 23 wherein said means for creating further comprises:

means for specifying a first at least one pathname for system program files;

means for specifying a second at least one pathname for application program files;

means for executing said main program unit using said first at least one pathname and said second at least one pathname for dynamically loaded program files; and

means for storing each application program file loaded during execution of said main program unit to said library file.

38. (new) The apparatus of claim 37 wherein said means for storing further comprises:

means for loading a program file when referenced during execution of said main program unit;

a1 Cont.

means for storing said program file to said library file when said program file is an application program file; and

means for determining whether execution of said main program unit has terminated.

39. (new) The apparatus of claim 38 wherein said library file further comprises a compressed file.

40. (new) The apparatus of claim 23 wherein said program files comprise class files and archive files.

41. (new) The apparatus of claim 40 wherein said system program file input comprises a collection of services and libraries included in an execution environment.

42. (new) The apparatus of claim 41 wherein said library file comprises an archive file.